AI-Based Conference Program Scheduler

Rucha Deshpande (034780087)
Purdue University
West Lafayette, Indiana, USA
deshpa37@purdue.edu

Aishwarya Devi Akila Pandian (34780639)
Purdue University
West Lafayette, Indiana, USA
aakilapa@purdue.edu

Vigneshwaran Dharmalingam (34780119)
Purdue University
West Lafayette, Indiana, USA
vdharmal@purdue.edu

ABSTRACT

Program creation is the process of allocating presentation slots for each paper accepted to a conference with parallel sessions. This process, generally done manually, involves intricate decision-making to align with multiple constraints and optimize goals. We propose an AI-based Conference Program Scheduler, a tool that uses AI-based techniques to automate the conference schedule creation process while addressing given constraints and maximizing session theme coherence. This project is motivated by the necessity to streamline the conference scheduling process, aiming to reduce the substantial manual effort currently required by program committee chairs [1]. We plan to use an optimization algorithm, the genetic algorithm to allocate sessions effectively. We plan to compare our results with the previous conference schedule data to evaluate the effectiveness and efficiency of the AI-based Conference Program Scheduler. Our evaluation will focus on the algorithm's ability to meet the outlined constraints and its success in creating thematically coherent sessions using metrics like constraint violation percentage and thematic coherence score.

CCS CONCEPTS

• **Computing methodologies → Artificial intelligence, Genetic algorithms, Natural language processing**; • **Theory of computation → Evolutionary algorithms**.

KEYWORDS

Automated Scheduling, Genetic Algorithms, Optimization Algorithms

ACM Reference Format:
Rucha Deshpande (034780087), Aishwarya Devi Akila Pandian (34780639), and Vigneshwaran Dharmalingam (34780119). 2024. AI-Based Conference Program Scheduler. In . ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

In the realm of academic and professional conferences, scheduling paper presentations is an important and complex task. Program Chairs have to manually allocate presentation slots while taking care of different constraints, including thematic alignment, speaker availability, and audience interest. Traditional manual scheduling methods are time-consuming, error-prone, and often fail to optimize for participant engagement. This can lead to suboptimal session timings, thematic mismatches, and logistical conflicts. The need for an automated scheduling tool arises from the requirement to overcome the limitations of manual scheduling. We propose an AI-based Conference Program Scheduler to automate the conference schedule creation process while addressing the given constraints. The constraints that make the creation of a program challenging are:

(1) Total time for all presentations in a session cannot be longer than the length of the session.
(2) Total number of sessions has to be equal to the number of sessions provided by the Program Committee chairs.
(3) If there are parallel tracks, then no two papers with common authors can be scheduled in parallel at the same time.

Another important consideration is the optimization goal: the papers in a session should be on a similar topic to avoid the parallel scheduling of sessions with related topics. The goal of our project is to reduce the manual effort required and minimize errors by automating this program creation process. After obtaining the topic-wise grouped papers, the next phase involves constructing the actual conference schedule, which adheres to the provided constraints (session lengths, number of sessions, parallel track limitations). Most scheduling problems, especially those with multiple constraints and optimization criteria, fall into the category of NP-hard problems [11]. Adaptive heuristic algorithms such as the Genetic Algorithm [7] are suitable for this type of scheduling problem as they often find good solutions in a reasonable amount of time. Genetic Algorithms are a type of evolutionary algorithm and optimization technique inspired by the process of natural selection. They iterate over generations of solutions to find the best or most fit solution according to a defined fitness function. Genetic Algorithms can optimize the overall schedule considering constraints and objectives [3]. Many related studies investigate the application of machine learning techniques for dynamic scheduling. Substituting ML techniques with Genetic Algorithms could potentially mitigate several challenges highlighted in the research. One of the limitations is the significant amount of data required for the machine learning models to converge to an effective solution. GAs do not require huge training datasets to find a satisfactory solution because they can explore a wide solution space more broadly and discover good solutions with less interaction with data. Moreover, the adaptive nature of GAs allows for periodic adjustments to the scheduling policy. Hence, we plan to apply the Genetic Algorithm

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/1122445.1122456

for our AI-based Conference Program Scheduler. We plan to assess the effectiveness of our approach using real-world data from the MSR conferences between 2018 and 2023, focusing on meeting scheduling constraints and achieving thematic session coherence. The evaluation will employ quantitative metrics: Constraint Satisfaction, Thematic Coherence, and Schedule Efficiency, to provide a comprehensive analysis of the tool's performance in solving the conference scheduling problem.

2 RELATED WORK

Conference scheduling is an intricate task that has seen various innovative approaches to optimize session organization. Riquelme et al. [9] explored a track-based scheduling method using simulated annealing to organize talks into thematic tracks. This heuristic approach aligns with our goal of solving complex scheduling challenges. Yet, our GA-based solution expands on this by integrating a consideration of various constraints, providing a comprehensive framework for conference planning.

Škvorc et al. [10] introduced NeSyChair, a system blending neurosymbolic paper representations with constrained clustering algorithms to enhance thematic coherence within sessions. This underscores the trend towards automating the scheduling process, aligning with our goals. However, our approach with GA aims for a balance between constraint satisfaction and thematic coherence, underscoring the dynamic nature of the optimization involved in conference scheduling.

The work by Manda et al. [8] offers an insightful approach through the utilization of heuristic optimization and topic modeling aimed at reducing thematic conflicts and improving schedule coherence. Their system, designed to automate and refine the conference scheduling process, demonstrates the potential benefits of AI-based scheduling solutions in addressing common scheduling challenges. While Manda et al. covers foundational concepts useful to our research, our AI-based scheduling tool introduces significant advancements in terms of adaptability and comprehensive constraint management. Our methodology extends beyond the scope of thematic alignment to encompass a broader range of logistical considerations.

The work of Bulhões et al. [2] on clustering-based conference scheduling introduces integer linear programming to optimize session alignment and enhance participant engagement. While our methodology emphasizes the use of genetic algorithm for its flexibility and efficiency in exploring vast solution spaces, the work of Bulhões et al. contributes significantly to our understanding of the theoretical and practical aspects of the scheduling challenge. Their approach to clustering based on common topics and subsequent optimization via integer programming offers a comprehensive framework for maximizing session benefits, a concept that complements our pursuit of thematic coherence and logistical adherence. By examining the strengths and limitations of their clustering-based strategy, we can better position our genetic algorithm-based solution within the broader context of AI-assisted conference scheduling.

Vangerven et al. [12] highlighted an attendee-centric model utilizing integer programming to reduce session-hopping and maximize participation, laying a foundation that emphasizes user experience in conference scheduling. Despite its efficiency in addressing attendee preferences, the model falls short on scalability and does not encompass the full breadth of conference complexities like thematic coherence, areas where our Genetic Algorithm approach could provide a more holistic solution.

Gündoğan and Kaya [5] presented an article similarity-based approach utilizing clustering methods like SBERT and SPECTER for organizing sessions, focusing primarily on textual similarity for thematic alignment. While their method provides content coherence, our GA-based strategy encompasses a broader perspective, addressing logistical intricacies alongside thematic alignment, presenting a more rounded approach to conference scheduling.

Herman et al. [6] discussed the application of GAs in academic scheduling, showcasing the adaptability of GAs in solving scheduling problems across different contexts. Although focusing on academic timetables, the principles and challenges they navigate parallel those in our conference scheduling context, demonstrating the versatility and efficacy of GAs in complex scheduling environments.

Lastly, Chilton et al. [4] introduced Frenzy, a collaborative platform for session organization, which contrasts with our algorithm-driven approach by emphasizing user collaboration and feedback. While Frenzy harnesses collective effort for data organization, our study leverages a GA-based model to automatically generate optimized schedules, minimizing manual intervention while ensuring comprehensive constraint satisfaction.

Together, these studies contribute diverse perspectives and methodologies to the conference scheduling domain. Our research builds upon these foundations, introducing a GA-based solution that respects logistical constraints as well as thematic coherence while aiming for optimal usage of available resources.

3 APPROACH

The input data sources for the challenge comprise the number of conference sessions, the length of each session, the number of parallel tracks, and the accepted papers in a particular conference along with all the corresponding data: title, topics, abstract, paper, authors and allocated time for the presentation of each paper. Our approach is specifically tailored to adhere to the constraints of conference scheduling—managing session lengths, ensuring no parallel scheduling of papers with common authors, and keeping the total number of sessions equal to the number provided by the PC chairs. Applying the Genetic Algorithm to the conference scheduling problem involves several key components. Below, we detail each component of our GA-based approach, as instantiated in our implementation:

3.1 Representation (Encoding)

The fundamental unit of our GA is the representation of potential conference schedules. Each schedule is conceptualized as a "solution" or individual in the GA's population. We encode a schedule as a collection of 'Session' objects, each comprising the given number of parallel tracks. A track is a list of 'Paper' objects, representing papers assigned to that track. This hierarchical structure allows us to model the complexity of conference schedules, including parallel sessions and varying session lengths. The 'Paper' class includes attributes such as 'id', 'authors', 'duration', and 'topic', which are

Time
9:00-9:30
9:30-10:00
10:00-10:30
10:30-11:00
11:00-11:30
11:30-12:00
12:00-12:30
12:30-1:00
1:00-1:30
1:30-2:00
2:00-2:30
2:30-3:00
3:00-3:30
3:30-4:00
4:00-4:30

Day 1

Session 1

Track 1	Track 2	Track 3
Paper 1	Paper 2	Paper 3
Paper 4	Paper 5	Paper 6
Paper 7	Paper 8	Paper 9
Paper 10	Paper 11	Paper 12

Session 2

Track 1	Track 2
Paper 13	Paper 14
Paper 15	Paper 16
Paper 17	Paper 18
Paper 19	Paper 20
Paper 21	Paper 22
Paper 23	Paper 24

Session 3

Track 1	Track 2	Track 3
Paper 27	Paper 28	Paper 29
Paper 30	Paper 31	Paper 32

Day 2

Session 4

Track 1	Track 2
Paper 33	Paper 34
Paper 35	Paper 36
Paper 37	Paper 38
Paper 39	Paper 40
Paper 41	Paper 42
Paper 43	Paper 44
Paper 45	Paper 46

Session 5

Track 1	Track 2	Track 3
Paper 47	Paper 48	Paper 49
Paper 50	Paper 51	Paper 52
Paper 53	Paper 54	Paper 55
Paper 56	Paper 57	Paper 58
Paper 59	Paper 60	Paper 61

Session 6

Track 1	Track 2	Track 3	Track 4
Paper 62	Paper 63	Paper 64	Paper 65

Day 3

Session 7

Track 1	Track 2	Track 3
Paper 66	Paper 67	Paper 68
Paper 69	Paper 70	Paper 71
Paper 72	Paper 73	Paper 74
Paper 75	Paper 76	Paper 77

Session 8

Track 1	Track 2
Paper 78	Paper 79
Paper 80	Paper 81
Paper 82	Paper 83
Paper 84	Paper 85
Paper 86	Paper 87
Paper 88	Paper 89

Session 9

Track 1	Track 2	Track 3
Paper 90	Paper 91	Paper 92
Paper 93	Paper 94	Paper 95

Day 4

Session 10

Track 1	Track 2	Track 3
Paper 96	Paper 97	Paper 98
Paper 99	Paper 100	Paper 101
Paper 102	Paper 103	Paper 104
Paper 105	Paper 106	Paper 107
Paper 108	Paper 109	Paper 110
Paper 111	Paper 112	Paper 113
Paper 114	Paper 115	Paper 116

Session 11

Track 1	Track 2	Track 3	Track 4
Paper 117	Paper 118	Paper 119	Paper 120
Paper 121	Paper 122	Paper 123	Paper 124
Paper 125	Paper 126	Paper 127	Paper 128
Paper 129	Paper 130	Paper 131	Paper 132

Session 12

Track 1
Paper 133
Paper 134

Figure 1: A sample schedule for a 4-day conference with 12 sessions of varying lengths and a different number of parallel tracks in each session. The papers in each track are shown to represent the overall structure of the schedule, but their durations need not be the same. The grey areas represent the breaks in between the sessions.

necessary for satisfying the scheduling constraints. Fig. 1 shows how a schedule for a 4-day conference with 12 sessions looks like.

3.2 Fitness Function

The fitness function is crucial in evaluating the viability of each schedule. It quantifies how well a schedule meets the predefined constraints and objectives, returning a fitness score that guides the selection process. In our implementation, the fitness function considers five types of penalties:

(1) **Time Penalty:** This component penalizes solutions where the total duration of papers in any track exceeds the maximum allowed session length. It ensures that all sessions fit within their allocated time slots, adhering to the conference's overall schedule.
(2) **Author Penalty:** To prevent scheduling conflicts, this penalty is applied when the same author is scheduled to present in parallel tracks within the same session. This component is crucial for respecting authors' availability and ensuring that attendees interested in papers by the same author do not face scheduling conflicts.
(3) **Distribution Penalty:** This penalty addresses the requirement that all accepted papers must be scheduled. It penalizes solutions that fail to include every paper in the conference program, ensuring comprehensive coverage of accepted submissions.
(4) **Dissimilarity Penalty:** This penalty is introduced to optimize the thematic coherence of papers scheduled in the same track of a session. It assesses the degree of similarity between the topics of papers in the tracks, applying a penalty when tracks diverge in their topics. The calculation of this penalty involves evaluating the overlap of keywords or topics associated with each paper, utilizing the Jaccard index for quantifying similarity. This component encourages the algorithm to schedule papers with related topics in the same session, enhancing the thematic flow of the conference program.

Caclulation of this penalty can be done as follows: Given two papers, A and B, with their topics represented as sets of keywords T_A and T_B, the Jaccard similarity coefficient, J, is calculated as follows:

$$J(T_A, T_B) = \frac{|T_A \cap T_B|}{|T_A \cup T_B|}$$

Where:

- $|T_A \cap T_B|$ is the number of keywords that appear in both T_A and T_B.
- $|T_A \cup T_B|$ is the total number of unique keywords in both T_A and T_B.
- $J(T_A, T_B)$ ranges from 0 to 1, where 0 means no similarity (completely different topics) and 1 means identical topics.

After calculating the Jaccard similarity for all unique pairs of papers in the track, aggregate these similarities to get a measure of overall thematic cohesion using average similarity. For a track with m unique pairs of papers,

$$S_{track} = \frac{1}{m} \sum_{i=1}^{m} J_i$$

Dissimilarity can the be given by,

$$D_{track} = 1 - S_{track}$$

The calculated overall track dissimilarity D_{track} can then be used to adjust the Dissimilarity Penalty in the fitness function.

(5) **Utilization Penalty:** Although not explicitly mentioned in the problem statement of the challenge, a potential additional penalty could be applied for significantly under-utilizing the available session time, leading to inefficient scheduling. This penalty is added as the amount of unutilized time for every session.

All the individual penalties obtained above will be normalized to $[0, 1]$ scale. The total penalty is a weighted sum of the individual penalties:

$$\begin{aligned}\text{Total Penalty} = w_1 \cdot \text{Time Penalty} + w_2 \cdot \text{Author Penalty} \\ + w_3 \cdot \text{Distribution Penalty} \\ + w_4 \cdot \text{Dissimilarity Penalty} + w_5 \cdot \text{Utilization Penalty}\end{aligned}$$

We initialize the weights in the following manner:

$$w1 = w2 = w3 > w4 > w5$$

This allows us to prioritize the three main constraints and then optimize for topic similarity followed by time utilization.

The Fitness Score is inversely related to the Total Penalty, meaning lower penalties result in higher fitness scores.

$$\text{Fitness Score} = \frac{1}{1 + \text{Total Penalty}}$$

This fitness function allows us to balance between strict adherence to constraints and optimization goals, guiding the GA towards feasible and coherent schedules.

3.3 Genetic Operations

3.3.1 Selection. We implement a selection mechanism to choose parent solutions for producing the next generation. Our selection process prioritizes solutions with higher fitness scores, enhancing the likelihood of inheriting desirable characteristics. We implement Roulette Wheel selection [13] to favor individuals with high fitness scores as well as strike the balance between the exploitation of the best solutions and exploration of the solution space.

3.3.2 Crossover. Crossover, or recombination, is a genetic operation that combines parts of two parent solutions to generate offspring. This process encourages the exchange of beneficial traits between solutions. In our system, the crossover operation involves swapping session slots between two schedules. We implement k-point crossover and experiment with different values of k. This operation is crucial for introducing diversity into the population and facilitating the exploration of new regions in the solution space. The effectiveness of crossover is partly controlled by the crossover probability—a parameter that determines how frequently this operation is applied to the population. A high crossover probability means that most of the offspring will be produced through recombination, promoting diversity in the population but also risking the disruption of good solutions. Conversely, a low crossover probability favors the retention of existing genetic material, which can slow down the exploration of the solution space. The optimal crossover probability balances these considerations, ensuring that the population evolves effectively towards an optimal schedule.

3.3.3 Mutation. Mutation introduces random modifications to a solution, preventing premature convergence to local optima and maintaining genetic diversity within the population. In our genetic algorithm, mutation randomly alters parts of a given schedule to explore new configurations that might not be reachable through selection and crossover alone. The mutation operation is applied with a certain probability, which is a parameter of the GA. This probability is chosen to balance the introduction of new genetic material with the preservation of good solutions already present in the population. Our approach includes two mutation strategies:

(1) **Swap Mutation:** This method involves selecting two papers within a session and swapping their parallel tracks. The swap mutation is designed to test the impact of different track assignments on the thematic coherence and the overall fitness of the schedule, without altering the set of papers within a session. It is a targeted approach that fine-tunes the schedule by exploring minor adjustments.
(2) **Move Mutation:** This approach allows the genetic algorithm to explore adjustments to the schedule by relocating a single paper from its current session to a different session. This type of mutation is mainly significant to explore the solution space for better time utilization.

3.4 Evaluation and Selection of the Next Generation

Post-crossover and mutation, we evaluate the fitness of the newly formed offspring. This evaluation process determines which solutions are retained for the next generation. Our methodology aims to select a high-quality set of solutions. We again implement Roulette Wheel Selection where solutions are selected based on probabilities proportional to their fitness scores, giving higher-scoring solutions a better chance of being chosen but still allowing lower-scoring solutions a chance to contribute to genetic diversity.

3.5 Termination

Our GA iterates through generations of solutions until it meets a termination condition. This condition is one of the following: reaching a maximum number of generations, achieving a fitness threshold, or observing negligible improvement over several generations. Termination criteria ensure that the algorithm concludes once it has sufficiently explored the solution space or found an optimally satisfying solution.

Fig. 2 shows the flowchart of the Genetic Algorithm for Conference Scheduling.

4 EVALUATION PLAN

To validate the effectiveness of our tool, we plan to conduct an evaluation using real-world conference data from the MSR 2018 to MSR 2023 programs. This will involve testing the scheduler's ability to meet the scheduling constraints outlined by the challenge and its success in creating thematically coherent sessions. To ensure a comprehensive evaluation of the conference scheduling problem, we plan the quantitative evaluation using the following metrics:

(1) **Constraint Satisfaction:** This metric assesses how well the scheduling algorithm satisfies the logistical constraints,

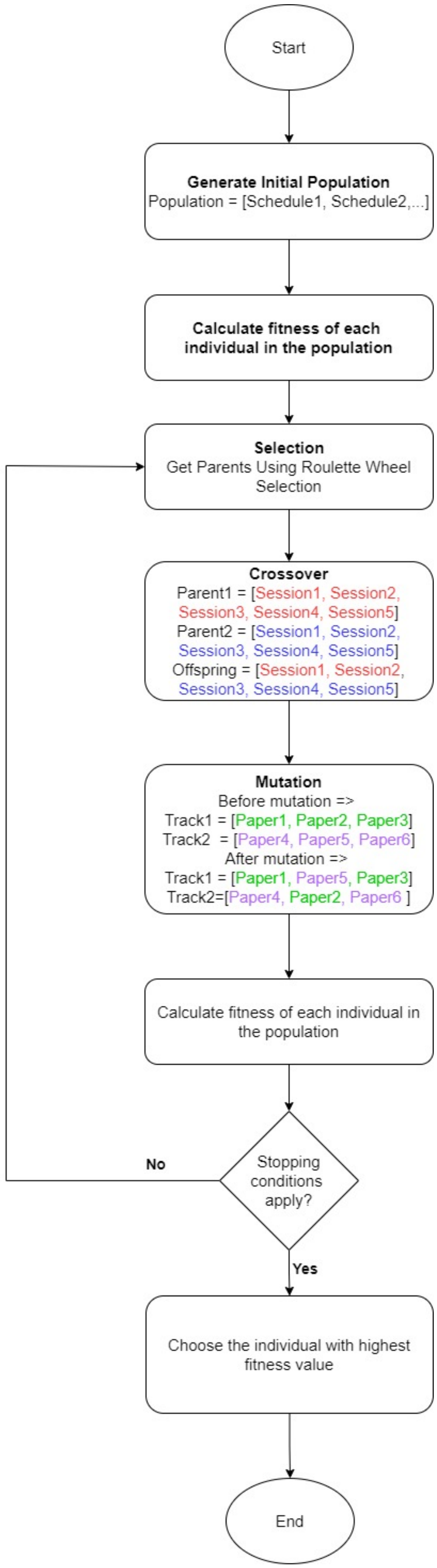

Figure 2: A flowchart showing the working Genetic Algorithm for Conference Scheduling.

including the number of sessions, session duration limits, and preventing scheduling conflicts among authors. The calculation will involve the percentage of sessions exceeding the time limit and the percentage of authors being scheduled in parallel tracks.

(2) **Thematic Coherence:** Thematic coherence measures how closely related the papers within the same session are, based on their topics. The calculation of similarity will be the same as the calculation outlined in Section 3.2.4.

(3) **Schedule Efficiency:** Schedule efficiency evaluates how well the conference schedule utilizes session capacities. The calculation of this metric involves the proportion of the total available conference time that is filled with presentations, aiming for high utilization without exceeding limits.

The evaluation of all the above metrics will give a holistic view of the algorithm's performance. These metrics have been chosen for their direct comparability to actual conference schedules, ensuring that our analysis is both grounded and actionable. By quantitatively assessing constraint satisfaction, thematic coherence, and schedule efficiency, we can identify specific strengths and areas for improvement in the scheduling algorithm.

REFERENCES

[1] AIware 2024. n.d.. AIware 2024 - Challenge Track - AIware 2024. https://2024.aiwareconf.org/track/aiware-2024-challenge-track Accessed on March 24, 2024.

[2] Teobaldo Bulhões, Rubens Correia, and Anand Subramanian. 2022. Conference scheduling: A clustering-based approach. *European Journal of Operational Research* 297, 1 (2022), 15–26.

[3] Edmund K Burke, David Elliman, and Rupert Weare. 1994. A genetic algorithm based university timetabling system. In *Proceedings of the 2nd east-west international conference on computer technologies in education*, Vol. 1. Citeseer, 35–40.

[4] Lydia B Chilton, Juho Kim, Paul André, Felicia Cordeiro, James A Landay, Daniel S Weld, Steven P Dow, Robert C Miller, and Haoqi Zhang. 2014. Frenzy: collaborative data organization for creating conference sessions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1255–1264.

[5] Esra Gündoğan and Mehmet Kaya. 2023. An Article Similarity-Based Approach for Planning Conference Sessions. In *2023 13th International Conference on Advanced Computer Information Technologies (ACIT)*. IEEE, 591–594.

[6] LSDL Herman and D Lantara. 2019. Lecture scheduling automation using genetic algorithm. In *East Indonesia Conference on Computer and Information Technology (EIConCIT), pp: 23*, Vol. 27.

[7] John H Holland. 1992. Genetic algorithms. *Scientific american* 267, 1 (1992), 66–73.

[8] Prashanti Manda, Alexander Hahn, Katherine Beekman, and Todd J Vision. 2019. Avoiding "conflicts of interest": a computational approach to scheduling parallel conference tracks and its human evaluation. *PeerJ Computer Science* 5 (2019), e234.

[9] Fabian Riquelme, Elizabeth Montero, Leslie Pérez-Cáceres, and Nicolás Rojas-Morales. 2022. A Track-Based Conference Scheduling Problem. *Mathematics* 10, 21 (2022), 3976.

[10] Tadej Škvorc, Nada Lavrač, and Marko Robnik-Šikonja. 2022. NeSyChair: automatic conference scheduling combining neuro-symbolic representations and constrained clustering. *IEEE Access* 10 (2022), 10880–10897.

[11] Bart Vangerven, Annette M.C. Ficker, Dries R. Goossens, Ward Passchyn, Frits C.R. Spieksma, and Gerhard J. Woeginger. 2018. Conference scheduling — A personalized approach. *Omega* 81 (2018), 38–47. https://doi.org/10.1016/j.omega.2017.09.007

[12] Bart Vangerven, Annette MC Ficker, Dries R Goossens, Ward Passchyn, Frits CR Spieksma, and Gerhard J Woeginger. 2018. Conference scheduling—a personalized approach. *Omega* 81 (2018), 38–47.

[13] Jinghui Zhong, Xiaomin Hu, Jun Zhang, and Min Gu. 2005. Comparison of performance between different selection strategies on simple genetic algorithms. In *International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC'06)*, Vol. 2. IEEE, 1115–1121.